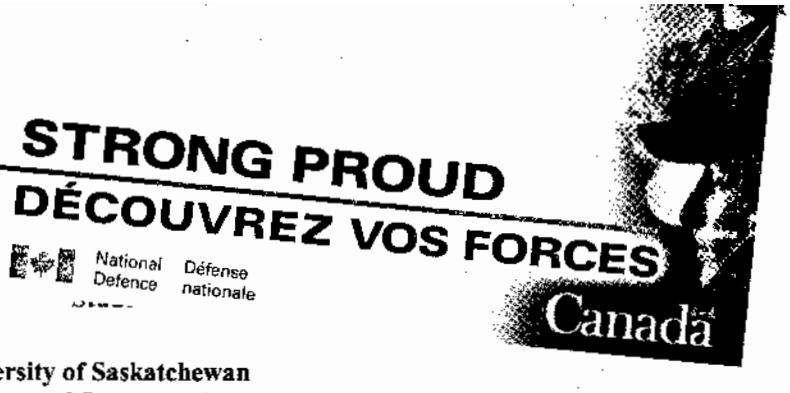


70
66%

University of Saskatchewan
Department of Computer Science

CMPT 374

Midterm– closed book / open mind
February 16, 00

Total Marks: 50 Time: 50 Minutes

Answer all of the questions in the spaces provided in this exam paper. If you don't have enough space, write on the back of the page, indicating clearly that your answer is continued there. Be sure to pace yourself according to the marks allotted to each question ... good luck!!!

A	10
B	6
C	5
D	4
E	8
Total	

17

Part A) SQL ***(10 Points)***

Please use the following relations when answering the questions of part A.

Relation:
P

PN	PName	Color	Weight	City
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

Relation:
S

SN	SName	Status	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Please write the result (relation) of the following SQL statements in the free space below each query. Each question is worth 2 points !

1)

```
SELECT SName, Status
FROM S
WHERE City = (SELECT City FROM P WHERE Color = 'Green')
```

SName Status

Jones 10

Blake 30

2)

```
SELECT COUNT(*) AS Places, City
FROM S
GROUP BY City
```

Places City

2 London

2 Paris

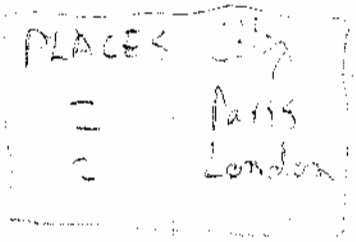
1 Athens

3)

```
SELECT DISTINCT Status
FROM S
```

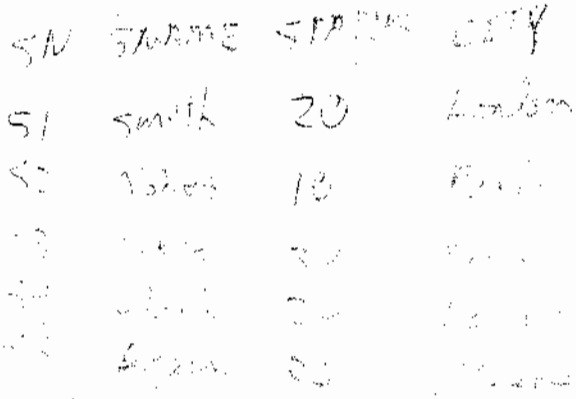
4)

```
SELECT COUNT(*) AS PLACES, City
FROM S
GROUP BY City
HAVING COUNT(*) > 1
ORDER BY City DESC
```



5)

```
SELECT *
FROM S
WHERE City < 'Sofia'
```



Part B) Relational Algebra *(10 Points)*

4 Points

1) **Name Codd's original eight algebraic operations.**

Restriction/Selection ✓
Union ✓
Join (natural) ✓
Cartesian Product ✓
Divide by ✓
Intersection ✓
Difference ✓
Projection ✓

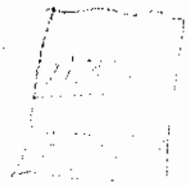
8

4 Points

2) **Some of Codd's original eight algebraic operations are considered to be primitive. Name the "non primitive" algebraic operations and redefine one of them by use of the original primitive algebraic operations.**

2 Points

3) **Show how to modify (update) a tuple by use of Codd's original eight algebraic operators.**



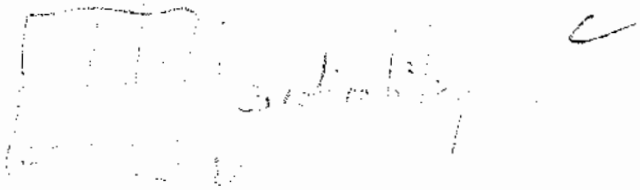
Part C) Basic Definitions *(10 Points)*

1 Point

1) **What is the meant with the "degree" of a relation?**

1 Point

2) **What is meant with the "cardinality" of a relation?**



2 Point

3) **Name the two types of data independence.**

a physical
b logical

2 Point → WANTED PICTURE

4) **Draw and explain the ANSI-SPARC three level architecture.**

PICTURE = 1point
EXPLANATION = 1point

(1)

External
- provides the users view interface

Conceptual
- provides a community view and defines schema

Internal
- describes physical level
- how data is physically stored

2 Points

5) **Name the four types of operations supported by the DML.**

~~DML = Data Manipulation Language~~

Insertion
Deletion
Update
Selection

✓

2 Points

6) **What is the difference between a "view" and a "base relation"?**

- Base relation is [illegible] in the database

✓

- a View is [illegible] relation [illegible] a view [illegible] base relations

Part D) The three record based data models *(12 Points)*

Assume that there is a N:M relationship between students and classes e.g. a student can take multiple classes and a class is taken by multiple students. Show how such a N:M relation is modeled in the three record based data models.

3Points

1) **Show how the N:M student-class relation is treated in a relational data model.**

DRAW THE THREE TABLES.

3 Points

2) **Show how the N:M student-class relation is treated in a network data model.**

3 Points

3) **Show how the N:M student-class relation is treated in a hierarchical data model.**

- The model is designed to N:M relationships
- N:M can only be supported by using multiple relations with duplicate information

- MENTION POINTERS, REDUNDANCY.

SHOULD HAVE 1 OR 2 CRITERIA FOR EACH MODEL.

3 Points

4) **Compare the three record based data models (use a table). Name specific strengths and weaknesses of each model.**

Strength | Weakness

Relational | Supports M:M relations easily | slow
Supports recursion easily

Supports 1:M relations

Hierarchical | Supports 1:M relations | only supports [illegible]
[illegible]

2

A → B

Part E) Functional dependencies (8 Points)

5 Points

1) **Determine the irreducible set of functional dependencies for the following 5 functional dependencies. Please document every step in determining the irreducible set.**

A -> BC
B -> C
A -> B
AB -> C
AC -> D

1. A→BC ⇒ A→B, A→C decomposition

2. AC→D
A→C ⇒ A→AC ⇒ A→D augmentative, transitive

3. AB→C
A→B ⇒ A→AB ⇒ A→C augmentative, transitive

4. So far we have
A→B A→C
A→C A→D
B→C
A→B

5. remove duplicates
A→B
A→C
B→C
A→D

6. A→B, B→C ⇒ A→C
Therefore can remove A→C
Therefore final set is
A→B, B→C, A→D

1 Point

2) **Which attributes of a relation are functionally depended on the candidate key?** [illegible]

attributes [illegible] are determined by the candidate key.

1 Point

5) **When is a relation in "third normal form"?**

[illegible] that is in First and second normal form and [illegible] dependencies [illegible]

1 Point

6) **What is the purpose of normalization?**

[illegible]

To eliminate [illegible] [illegible] relation